KUNST
UNICON

Unidimensional conjoint measurement

By Maria Thissen

RTOG-FSW
Radboud University Nijmegen

September 2012

Content

1 General description

Unicon is a program for unidimensional conjoint measurement: an ordinal dependent variable is measured under two to five nominal condition variables.

If one tries to test if changes in a dependent variable can be accounted for by changes in some independent variables, one will usually first measure the variables involved and then analyze the relations between them. In order to do so, one needs measuring instruments for all the variables and a hypothetical functional relation between the independent variables and the dependent variable to be tested.

Many psychological theories predict a functional relation of some kind but there is no unique procedure for the measurement of the relevant variables. Often there are many different measures for the same property and they need not to be related by any simple relation (like a linear transformation).

Conjoint measurement tries to construct the measurement scales of the independent variables so that the data obey a predicted relation. If this attempt succeeds, one knows that there exist scales of measurement that satisfy the proposed composition rule and gets the scale values for free. Conjoint measurement was developed by Luce and Tukey (1964).

Unicon performs conjoint measurement with up to 5 independent variables and one single dependent variable. The independent variables are called factors. Each factor must consist of just a few categories or, as they are called in this context, *levels*.

The data consist of measurements on one or more cases, in this context called *replications*, and each replication contains one value of the dependent variable for each combination of levels of the factors. These combinations are called the *cells* of the replication. The dependent variable must be of ordinal level. The result of the analysis will be a scale value for each level of each factor. If one applies the hypothetical composition rule, values of the dependent variable are obtained that, to some extent, show the same order as the original data for each replication. Disturbances of the order may be attributed to random fluctuations in the data or to failure of the program to find the appropriate solution, but they may also lead to the conclusion that the composition rule has to be rejected.

Intensity	Duration					
	1	2	3	4	5	6
1	145	163	170	202	184	214
2	337	477	557	635	637	656
3	763	1035	1203	1355	1352	1595
4	1302	1435	1822	1900	1927	2195
5	1650	2072	2203	2545	2684	2781

Figure 1: One replication for a two-factor model.

Figure 1 shows a data matrix with two factors and one dependent value for each combination of levels. The factors are:

Duration = the duration of an electrical shock in seconds (reduced to 6 categories)
Intensit = the intensity of the shock (reduced to 5 categories)

The dependent variable measures the unpleasantness of the shocks. In this example each cell contains the mean score of a number of persons, but the researcher could as well have chosen to enter the individual persons as separate replications. *Unicon* uses only the **order** of the dependent values **within** each of the replications.

The program requires a composition rule, also designated as the *model* of the relation between the factors and the dependent variable, for instance: `Unpleasantness = Duration × Intensity`. If the data in figure 1 are analyzed with this model the results will be as shown in figure 2.

		Duration					
		0.8815	1.0037	1.1450	1.2439	1.2439	1.3463
Intensit		1	2	3	4	5	6
⇓							
-0.0013	1	-0.0011	-0.0013	-0.0015	-0.0016	-0.0016	-0.0017
0.8058	2	0.7103	0.8087	0.9226	1.0023	1.0022	1.0848
1.4151	3	1.2474	1.4203	1.6203	1.7601	1.7601	1.9051
1.8980	4	1.6731	1.905	2.1733	2.3609	2.3609	2.5553
2.4316	5	2.1435	2.4407	2.7843	3.0247	3.0246	3.2737

Figure 2: Unicon solution of the data in figure 1 using a multiplicative model.

Factor-levels		Data	Solution values	Factor-levels		Data	Solution values
1	1	145	-0.00114	4	1	1302	1.67307
1	2	163	-0.00130	3	5	1352	1.76013
1	3	170	-0.00148	3	4	1355	1.76015
1	5	184	-0.00161	4	2	1435	1.90505
1	4	202	-0.00161	3	6	1595	1.90507
1	6	214	-0.00174	5	1	1650	2.14346
2	1	337	0.71026	4	3	1822	2.17329
2	2	477	0.80874	4	4	1900	2.36088
2	3	557	0.92261	4	5	1927	2.36086
2	4	635	1.00225	5	2	2072	2.44066
2	5	637	1.00224	4	6	2195	2.55526
2	6	656	1.08477	5	3	2203	2.78431
3	1	763	1.24735	5	4	2545	3.02466
3	2	1035	1.42030	5	5	2684	3.02463
3	3	1203	1.62029	5	6	2781	3.27368

Figure 3: Data versus cell values from the *Unicon* solution

Figure 3 shows the values of the data in ascending order and the corresponding cell values according to the *Unicon* solution. The cell values turn out to have almost the same order as the original data, so the model fits the data quite well.

The first version of *Unicon* was developed by E.E.Ch.I. Roskam.

2 Conjoint measurement in detail

Consider a multi-factor design A × B × C × ..., where A, B, C… are factors such as experimental conditions.

We define:

- f = number of factors
- F_i = the i-th factor
- l_i = the number of levels in the i-th factor.
  In the example of figure 1, duration has six levels and intensity has five.
- X_r = The r-th replication: the number of elements in X_r is $\prod_{i=1}^{f} l_i$. In the example of figure 1, X_r contains 30 cells (the 30 cells within a single replication).
- $x_{rij}...$ = the observation in cel i,j,... for replication r
- $o_{rij}...$ = the ranking of observation $X_{rij}...$. within replication r
- s_{ij} = a scale value for level j of the i-th factor

The underlying model assumes that the data define a weak order that is equal to the order defined by a simple function of the scale values. This means that there exist the following values:

$z_{ij}... = t(s_{1i},s_{2j},...)$ such that: $o_{rij...} \geq o_{rkl...} \leftrightarrow z_{ij...} \geq z_{kl...}$ (1)

These z-values are also called the *dependent values*. They are the same for all replications.

In practice, it will seldom be possible to find scale values s that conform completely to (1). So values are sought that come as close as possible.

The function t is called the model. The simplest model is the additive one, stating that the value of the dependent variable is equal to the sum of the corresponding scale values:

$z_{ij}... = t(s_{1i},s_{2j},...) = s_{1i} + s_{2j} + ...$

2.1 Fitting values and stress

In order to measure the quality of a solution, intermediate values ($\hat{z}_{rij}$) are computed between the z-values ($z_{ij...}$) and the data (o_{rij}...). These intermediate values stay as close to the z-values as possible, but they are not allowed to violate the order in the data. So the differences between the z and the $\hat{z}$ embody the friction between the observed data and the model assumptions. The intermediate values are called *disparities* or *fitting values*. Figure 4 shows a plot of the rank order of the data (the x-axis) and the z-values as well as the fitting values (both on the y-axis) in a hypothetical configuration. The z-values are represented by the symbol | , the fitting values are shown as dashes (-). In a perfect solution, the z-values must increase from left to right or at least stay on the same level. In that case, the fitting values coincide with the data. A plot like the one in figure 4 is called a Shepard* plot.

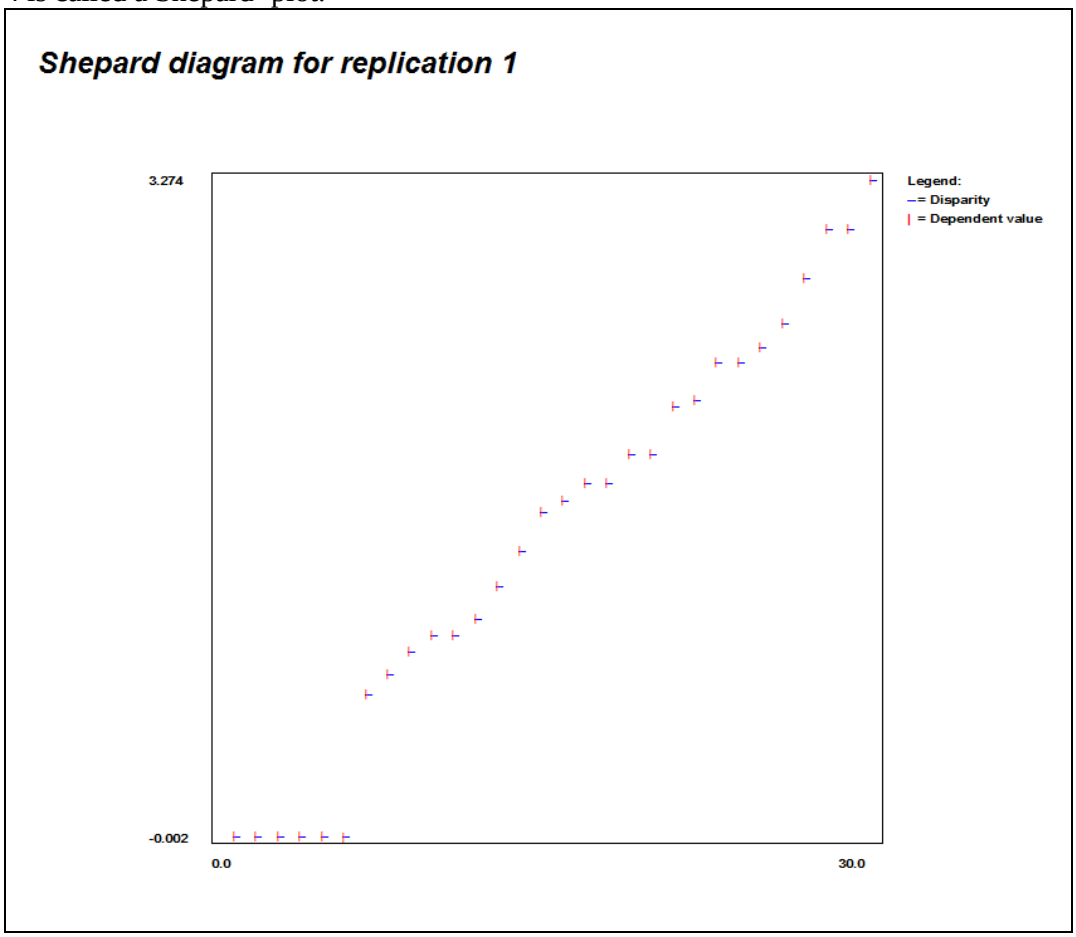

Figure 4: A Shepard-plot.

The program finds its solution by choosing a suitable trial configuration (an initial set of scale values) and then iteratively minimizing the stress or loss function *S-form2*, i.e. a quantity that indicates how well a given solution (a set of scale values) meets the requirement of monotonicity as defined by (1).

---

* After its inventor R.N. Shepard

$$\text{S-form2} = \frac{\sum_r \sum_i \sum_j \sum_{\ldots} (z_{ij\ldots} - \hat{z}_{ij\ldots r})^2 e_{ij\ldots r}}{\sum_i \sum_j \sum_{\ldots} n_{ij\ldots} (z_{ij\ldots} - \bar{z})^2}$$

$\hat{z}_{rij\ldots}$ = *disparity* or *fitting value* closest to $o_{ij}\ldots$ at the r-th replication. The $\hat{z}$ values are computed according to Kruskal's principle of monotone regression (see Kruskal, 1964a), which means that they have a least square fit to the z-values under restriction of weak monotony with the o-values. The value of $\hat{z}_{rij\ldots}$ is undefined when $e_{rij\ldots} = 0$, but that does not affect the value of S-form2.

$e_{rij\ldots}$ = 0 if $o_{r_{ij\ldots}}$ is missing

= 1 if $o_{r_{ij\ldots}}$ is known

$n_{ij\ldots} = \sum_{r=1}^{k} e_{rij}$ for all k replications.

and $\bar{z} = \frac{\sum_i \sum_j \sum_{\ldots} n_{ij\ldots} z_{ij\ldots}}{\sum_i \sum_j \sum_{\ldots} n_{ij\ldots}}$

The $\hat{z}$-values will be different for each replication, but the scale values and consequently the z-values are the same for all replications.

During the search for the best solution *Unicon* standardizes the z-values continuously through division by their standard deviation. The scale values are standardized accordingly. There are however models that do not allow such a standardization.

The stress of the solution is invariant under linear transformations of z, but not under other transformations. Small changes, however, which do not influence the rank order of the z-values, are allowed. Therefore, it can be said that the measurement approaches an interval level, but it does not reach it completely.

The algorithm does not guarantee that the best possible solution is found, since it may land in what is called a *local minimum*. For this reason *Unicon* allows the user to repeat the process one or more times. Such a repetition starts from the last configuration, but the iteration counter is set back to zero and the step-size, which determines the amount of change in the independent scale values, is set to 0.20 + 0.05r, where r is the number of restarts still to be performed. Since this is a large value, the change may be rather drastically; hence the phenomenon is called a *violent motion*.

If during the iteration process the stress becomes too high (greater than 0.99) a new configuration will be generated at a random base, up to 10 times. At each new start the iteration counter is set back to zero and the step-size will be the same as with a violent motion.

2.2 Trivial solutions

The requirement of weak monotonicity can always be met by setting all scale values (and thereby all z-values) such that the z-values are almost equal. Although this is counter-acted by the choice of the denominator in the formula S-form2, it may happen that a low stress is bought by making at least some z-values (almost) equal, even if the corresponding o-values are different. This phenomenon is called *trivialization* or *degeneration*. In the example of figure 2, for instance, the fourth and the fifth level of the factor *duration* have received almost the same scale value. The number of equivalence classes in the z-values, as compared to the number of equivalence classes in the o-values, is an indication of the degree of trivialization.

2.3 The models

Conjoint measurement analysis is performed on data consisting of ordinal relations between the cells within one or more replications, that is, data of the form:

$$x_{rij\ldots} \geq x_{rkl\ldots}$$

The empirical interpretation of the relation ≥ is to be defined by the experimenter. They may be, for example, preference data. It is also the task of the experimenter to decide by what means the occurrence of ≥ is assessed.

A simple way would be that a subject performs pair comparison judgments on all pairs of cells, using some criterion provided through the experimental instruction. For example, the subject must judge which of the level combinations (i,j,...) and (k,l,...) is the more attractive one.

The rank orders of the x-values within each of the replications constitute the essential data of the experiment. If the data are collected by way of pair comparisons, the experimenter must construct a full rank order for each replication. One way of doing so, is to take the vote count; that is: one counts how many times a combination (i,j,...) was judged or observed to be ≥ another combination (k,l,...). (This may not always be a good manner of obtaining a full order; discussion of this problem is beyond the scope of this manual.)

Alternative methods of obtaining a full rank order make use of balanced incomplete block designs, that is, the subject is presented with a triple, a quadruple (in general: an n-tuple of cells) of combinations and instructed to rank order these. This procedure is carried out for a set of n-tuples, such that every pair of combinations appears in the same number of n-tuples. The full rank order is then obtained by taking the vote count.

One can also use a single stimulus method, having the subject rate the stimuli on, say, a 7-point rating scale, according to whatever the criterion for the dependent variable is. Ties may occur.

Occasionally missing data may occur whenever any of the $x_{rij\ldots}$ is not known.

The transformation function $t(s_{1i},s_{2j},...)$ expresses the conjoint measurement model. In *Unicon* it must be a simple composition function linking the factors by means of the operators +, -, * or /, while using each factor only once.

The most commonly used models are:

Additive:	$s_{1i} + s_{2j} + ...$
Distributive:	$(s_{1i} + s_{2j}) \times s_{3k}$
Dual-distributive:	$(s_{1i} \times s_{2j}) + s_{3k}$
Multiplicative:	$s_{1i} \times s_{2j} \times ...$

The division operator should be avoided as much as possible, because the model function will be discontinuous at the points where the denominator is zero.

The program allows the restriction that some factors are completely or partly equivalent. This means that some levels of one factor must receive the same values as the corresponding levels of another factor. Such a restriction may be appropriate if, for instance, two factors refer to the same measurement instrument under two different conditions, for instance the loudness of a sound presented to the left ear (factor 1) and of one presented to the right ear (factor 2).

2.4 The trial configuration

The search for the best fitting scale values starts from a so-called trial configuration: a set of initial scale values, which will be adjusted in small steps in order to minimize the stress.

2.4.1 Automatically generated trial configuration

By default, the trial configuration will be generated automatically by the program. Two competing methods are used. Eventually the one with the lowest stress will be applied.

	Duration					
Intensity	3.85	5.85	7.45	9.05	9.25	11.05
-4.25	1	2	3	5	4	6
1.75	7	8	9	10	11	12
8.42	13	14	15	18	17	20
13.92	16	19	22	23	24	26
18.92	21	25	27	28	29	30

Figure 5: Ranks and first estimates of scale values for the data of figure 1.

Method 1

1. For each level of each factor, the scale value is set to the average rank of the cells belonging to that level, diminished by $\frac{f-1}{f}$ times the overall mean of the ranks (where f denotes the number of factors). Figure 5 shows the result of these calculations for the data in figure 1. This approach results in a least squares fit for the additive model under the restriction that the mean of the scale values is the same for all factors*. So it is especially appropriate if the data can be fitted by a purely additive model or by a multiplicative model with all positive scale values.

* Without such a restriction the model would be unidentifiable.

2. If some factors must be completely or partly equivalent, the following adjustments are made. Corresponding scale values are replaced by their average. However, if two equivalent factors have different numbers of levels, this adjustment would have as a result that the mean value of the z-values differs from that in the o-values.
   Assume that factors i and j must be equivalent and that $l_j > l_i$. Let us indicate the new values for s_{jk} with $k < l_i$ by the symbol s'_{jk}, so $s'_{jk} = \frac{s_{ik} + s_{jk}}{2}$.
   Define: $d_k = s'_{jk} - s_{jk;}$ then: $s'_{jk} = s_{jk} + d_k$ and $s'_{ik} = s_{ik} - d_k$
   We define also: p'_i = mean of the new scale values for factor i
   $p' = \sum_{i=1}^{f} p'_i$
   Now $p' = \sum_{i=1}^{f} \frac{1}{l_i} \sum_{j=1}^{l_i} s'_{ij} = p + \left(\frac{1}{l_j} - \frac{1}{l_i} \right) \sum_{k=1}^{l_i} d_k$
   In order to get the mean of the z-values equal to the mean of the o, we add the value $\frac{1}{l_i} \sum_{k=1}^{l_i} d_k$ to each of the values s_{jk} with $k > l_i$.
3. For each pair of scale values, say s_{ij} and s_{kl}, the following modifications are tested:
   (1) $s_{kl} \rightarrow -s_{kl}$,
   (2) $s_{kl} \rightarrow 1/s_{kl}$,
   (3) $s_{kl} \rightarrow -1/s_{klj}$,
   (4) $s_{ij} \rightarrow -s_{ij}$,
   (5) $s_{ij} \rightarrow -s_{ij}$ and $s_{kl} \rightarrow -s_{kl}$,
   (6) $s_{ij} \rightarrow -s_{ij}$ and $s_{kl} \rightarrow 1/s_{kl}$,
   (7) $s_{ij} \rightarrow -s_{ij}$ and $s_{kl} \rightarrow -1/s_{kl}$,
   (8) $s_{ij} \rightarrow 1/s_{ij}$
   (9) $s_{ij} \rightarrow 1/s_{ij}$ and $s_{kl} \rightarrow -s_{kl}$,
   (10) $s_{ij} \rightarrow 1/s_{ij}$ and $s_{kl} \rightarrow 1/s_{kl}$,
   (11) $s_{ij} \rightarrow 1/s_{ij}$ and $s_{kl} \rightarrow -1/s_{kl}$,
   (12) $s_{ij} \rightarrow -1/s_{ij}$
   (13) $s_{ij} \rightarrow -1/s_{ij}$ and $s_{kl} \rightarrow -s_{kl}$,
   (14) $s_{ij} \rightarrow -1/s_{ij}$ and $s_{kl} \rightarrow 1/s_{kl}$,
   (15) $s_{ij} \rightarrow -1/s_{ij}$ and $s_{kl} \rightarrow -1/s_{kl}$
   For each of the sixteen configurations from steps 1 and 2 the stress is computed and the configuration with the lowest stress is kept.
   If the number of scale values exceeds 50, this procedure is restricted to single scale values and pairs of scale values that belong to different factors.
4. The procedure in step 3 is repeated until no more improvement of the stress can be accomplished. There is however a maximum of 30 repetitions.

Method 2

This method is very similar to the first, but now the ranks are replaced by their logs and at the end the resulting scale values are replaced by their antilogs.

2.4.2 User defined trial configuration

Although the program can generate a trial configuration, there may be reasons for the user to define one himself:

- Maybe you are not satisfied with a solution from a previous *Unicon* run and you want to guide the program in a certain direction. Especially with complex models, like $\frac{a+b}{c-d}$, the generated trial configuration may be quite poor and lead to a local minimum for the stress.
- There may be theoretical grounds to expect some type of configuration to be the best.
- If several groups of replications are analyzed you may want to use the same trial configuration for all of them, or maybe you want to use the outcome of one analysis as a start for another.

When the user defines the trial configuration, but not all elements are filled, the program will generate a complete trial configuration and then fill the holes in the user given configuration with values from the generated configuration. Before doing so, the generated configuration is rescaled so that the sum of absolute scale values is equal to the corresponding sum in the given configuration. These sums are computed over those scale values that are known in both configurations.

2.5 Minimizing the stress

Unicon will adjust the configuration step by step in order to decrease the stress S-form2. To improve the configuration, the method of steepest descent is used. This is an iterative procedure. For each scale value s_{ij} (for level j of factor i) in the configuration the program computes the partial derivative of the stress (d'_{ij}), i.e. a value, such that the stress decreases most rapidly when the new configuration is adjusted by:

$$s_{ij(t+1)} = s_{ijt} - \alpha_t \, d'_{ij(t)}$$

with a small step size α; t indicates the iteration step. In practice, α is chosen according to the method described by Kruskal (1964a).

2.6 The treatment of ties

A tie in the data means that some of the observations receive the same rank order. Ties can be handled in two ways

"Break": means that tied data may receive different ranks.
"Let": tries to give tied data equal rank orders. Therefore data values within a tie are replaced by their average rank order.

Data cells are considered to be tied if they do not differ more than a certain amount ε, in the following sense:

Consider the four cells a, b, c and d with rank order a < b < c < d. If the difference between a and b is not greater than ε, the elements a and c are compared. If these do not differ more than ε, a will be compared to d. Suppose that a and d differ more than ε, than the elements a, b and c are tied, i.e. they receive the same rank number; d will not join the tie although it may differ less than ε from b or c.

3 The main algorithm

The main algorithm performs the following steps:

a The data within each replication are replaced by rank numbers.

b If the user does not provide a trial configuration, the program generates one.
If the user specifies an incomplete trial configuration, it is completed from the generated configuration.

c The z-values, the fitting values and the stress are computed according to the model.
If the stress is the best stress so far, the configuration is saved.

d Now it is decided if the process must go on and how. This decision depends on a number of criteria. These criteria are described in the next section (see 3.1).
If the iteration process must end, z-values, fitting values and stress are computed for the last time, based on the best solution found. Then the algorithm continues with step f.

e The configuration is adjusted by moving each point a small step such that the stress probably decreases (using partial derivatives of the stress with respect to the configuration).
If in one step the stress increases by more than 20% or increases to over 0.95, the algorithm will step back to its previous configuration and reduce its step size; this process of reducing the step size may be performed up to 5 times in succession. A new configuration will be formed using the new step size.
The configuration is standardized and the algorithm loops back to step c.

f At the end of the process the final configuration is standardized.

3.1 Evaluation of the stress

The decision whether the iteration process must continue or not depends on the following decision rules, evaluated in the given order:

- Stop when the stress is less then the criterion value given by the user.
- Stop when the stress is more or less stable, that is when the ratios of the 5 previous stresses with the last stress lay between 0.997 and 1.003.
- Stop when the maximum number of iterations has been reached.
- If the stress is less than the criterion value defined by the user, the process must continue with a number of additional iterations, up to a user defined maximum number, as long as the stress is going down (new stress $\leq$ previous stress - $0.005 \times$ criterion).
- If the process must stop, but the minimum stress is greater than the criterion value, a restart (violent motion) will be performed, if allowed by the user.
- Always stop when the stress is less than 0.00001.

4 Files

There are six file types that are important for this program:

data files:

The files that contain the data to be analyzed. One program run can analyze several files and each file can contain several replications.
There may also be a file with the trial configuration.

It is very important that the data are arranged properly. For each replication a full set of x-values must be given; the missing cells must be filled with a code for missing data. The number of cells in such a data matrix is the product of all numbers of levels. The maximum number of cells per replication is limited, depending on the computer system. At the moment this documentation is written, the maximum for the Windows version is 500.

There must be a correspondence between the order in which the factors are defined and their arrangement within a replication. In the data, the first factor must change its levels slower than the second, the second slower than the third and so on. Figure 6 shows a scheme for a model with three factors: A with two levels, B with three levels and C with five levels. A changes its levels after 15 values, B after 5 values and C after 1. Each replication must contain 2*3*5 = 30 cells.

$A_1B_1C_1$	$A_1B_1C_2$	$A_1B_1C_3$	$A_1B_1C_4$	$A_1B_1C_5$
$A_1B_2C_1$	$A_1B_2C_2$	$A_1B_2C_3$	$A_1B_2C_4$	$A_1B_2C_5$
$A_1B_3C_1$	$A_1B_3C_2$	$A_1B_3C_3$	$A_1B_3C_4$	$A_1B_3C_5$
$A_2B_1C_1$	$A_2B_1C_2$	$A_2B_1C_3$	$A_2B_1C_4$	$A_2B_1C_5$
$A_2B_2C_1$	$A_2B_2C_2$	$A_2B_2C_3$	$A_2B_2C_4$	$A_2B_2C_5$
$A_2B_2C_1$	$A_2B_3C_2$	$A_2B_3C_3$	$A_2B_3C_4$	$A_2B_3C_5$

Figure 6: Example of the layout of a data matrix (one replication).

trial configuration file:

If the user specifies his own trial configuration, he can do so in three ways:

1. By specifying the trial configuration before the data in the first (or only) data file to be analyzed. If there are several data files, this may be a bit tricky, since the order in which the files are analyzed may be different for different operating systems.
2. By specifying the trial configuration in a separate file.
3. By specifying the trial configuration interactively during the first program phase, in which the options are set. If the configuration is defined in this way and the settings are saved in a settings file (see below), the trial configuration will be saved with them.

Even if there are several data files, there can be only one user defined trial configuration, which will be used for all analyses.

If the trial configuration is given in a file, the file must contain a list with the scale values, ordered by factors and, within each factor, by level. Figure 7 shows an example of a trial configuration for the data of figure 1. The first five numbers specify the five scale values for the first factor (*intensity*), the last six the six scale values for the second factor (*duration*).

```
3.5 9.5 16.17 21.67 26.67 11.6 13.6 15.2 16.8 17.0 18.8
```

Figure 7: Example of the layout of a trial file.

settings files:

These files are used to save the options as they are specified by a user. A settings file contains all information about the analyses to be performed, including the description of the data and, possibly, the trial configuration, but not the data themselves. Many settings files may be available. By default, their extension is '`.setuni`'.

listing files:

A listing file contains the main results of an analysis in a nice layout for human readers. There will be one listing file for each data file. The name will borrow its first part from the data file and end on '`.LST`' (or on '`1.LST`', '`2.LST`', ... and so on). You can inspect it by any editor, but in order to have an orderly layout you must view it in a small non-proportional font like `Courier New 9`. Listing files are stored in the folder in which the corresponding input files are found.

raw output files:

Depending on the chosen options, the program may produce one file with the final configuration for each input file. These raw files are meant to be input to other programs, for instance as trial configurations for other *Unicon*-runs. Their names will take their first part from the data files and end with '`.OUT`' (or '`1.OUT`', '`2.OUT`', ... and so on).

plot files:

Depending on the chosen options, the program may produce one file with a Shepard plot for each data file. See 2.1 for an explanation. The names of these files will take their first part from the data files and, on Windows machines, they will end with '`S.BMP`', '`1S.BMP`', '`2S.BMP`' and so on.

5 Installing the program on Windows

The installation of the program is very simple:

1. Copy the file *Unicon.exe* to any place on your hard disk. Optionally you may make shortcuts on the task bar and/or the desktop.

2. After the first time you have used the program, double click on the listing file. Windows will ask you to select the program to be used when opening the file. Select a simple text editor like Notepad or WordPad.

3. After the first time you have saved the program settings, double click on the settings file. Windows will ask you to select the program to be used when opening the file. Select the program *Unicon.exe* or any shortcut to it.

That is all: from now on, you can start the program by double clicking the exe-file, one of its shortcuts or a settings file.

6 Running the program

To run *Unicon* you must double click on its executable file (for Windows: *Unicon.exe*) or, if you have used the program before, on one of its settings files (for instance *Current.SetUni*). The first thing you will see then is the main window, as shown in figure 8.

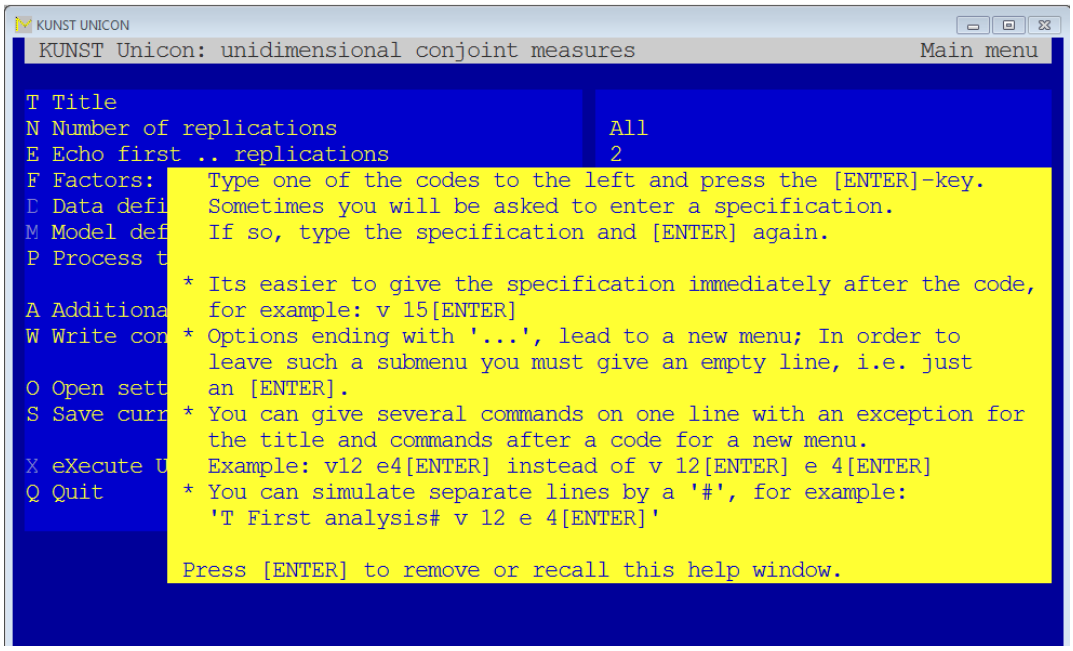

Figure 8: The main window.

On the screen the light part with the text '`Type one ...`' is a yellow text window. Windows like that contain hints and explanations. If you have read the text (or do not need it), you can press the Enter-key and the yellow window will vanish (see figure 9).

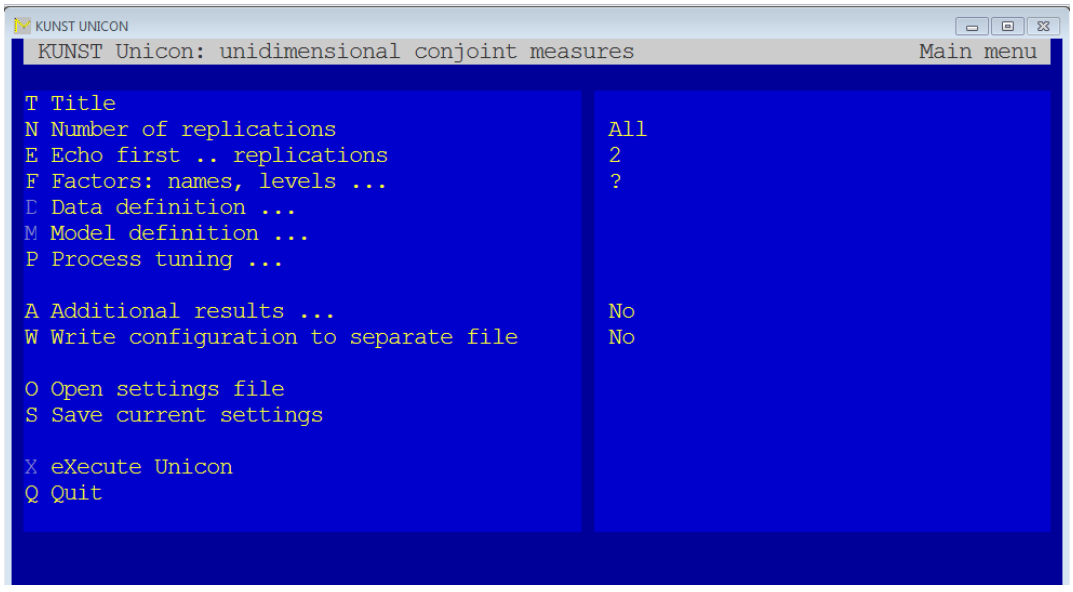

Figure 9: The main window without help-window.

Now you can see the entire main window. The left part is the main-menu. It consists of a list of options, each one preceded by a one-character code. To select an option, you must type the code, followed by the information you want to give and then the Enter-key. From now on, we will indicate the Enter-key as: `Enter`. You may for instance type: `T Analyzing first session`Enter to define the title that will appear as a header in the listing files. If you do not know what the meaning of an option is, you may just enter its code and `Enter`. There will appear a question on the screen and, if helpful, a yellow window to give you information. Press `Enter` to remove the yellow window and then give the specification belonging to the code, or ignore the yellow window and give the specification at once. **Do not repeat the code itself!**

The right part of the window gives a short review of the options, as they are currently set. In the example of figure 9, you can see the following:

- No title line is defined.
- The number of replications is derived from the data file(s).
- The data of the first two replications will be echoed in the listing file(s).
  If the trial configuration is read from a file, it will also be echoed in the listing file(s).
- The factors are not yet defined, but must be specified.
- The data definition-menu cannot be chosen before the factors are defined.
  (Although it may be difficult to see in this print, the option character '`D`' is gray).
- The model cannot be defined before the factors are defined.
  (Therefore the option character '`M`' is gray).
- The process tuning options are not echoed here, but in the corresponding submenu.
- No additional results will be reported.
- No raw output file will be produced.

If a menu option end with three periods (...) the option leads to a submenu.

7 Menu options

7.1 The main menu

The *main* menu (see figure 9) contains the following options:

7.1.1 T Title

This option allows you to specify a header to be used in the listing files.

7.1.2 N Number of replications

By this option, you may specify how many replications are to be analyzed together. This option holds for each of the data files. The default value is 'all', that is: all replications in the file have to take part in the analysis.

7.1.3 E Echo first .. replications

By typing `e ##Enter` you specify that the first ## replications of each data file must be shown in the corresponding listing file. This may help you to check if the input specifications and the input data match correctly. The default value for this option is 2. If the trial configuration is given in a file and the echo-value is greater than zero, the trial configuration will also be echoed in the listing file(s).

7.1.4 F Factors: names, levels ...

If you type `FEnter` you are asked for the number of factors, unless this number is already known. Then the factors-menu will appear, where you can define the names and the levels of the factors and information about equivalences. This menu will be discussed in 7.2. The factor levels must be given before you can specify the model and the data.

7.1.5 D Data definition ...

If you type `DEnter` the main-menu will be replaced by the data-definition-menu. This menu allows you to give information about the input data and the trial configuration. It will be discussed in 7.3.

7.1.6 M Model definition ...

This option leads to the model-definition-menu, in which you can specify the model. It will be treated in 7.4.

7.1.7 P Process tuning ...

This option leads to a new menu where you can specify some options that influence the process, like the number of iterations, the minimum stress and the handling of ties. It will be treated in 7.5.

7.1.8 A Additional results ...

This option leads to a new menu that allows you to order additional information in the listing file. It will be treated in 7.6.

7.1.9 W Write configuration to separate file

By this option you can specify whether the final configuration must be stored in a “raw” text file or not.

7.1.10 O Open settings file

If you have ever saved a set of options for *Unicon* or if you have received a settings file from someone else, you can retrieve the options from the settings file. If you type `O`Enter, a file-selector box will appear on the screen that allows you to select the settings file. Figure 10 shows such a file selector box.

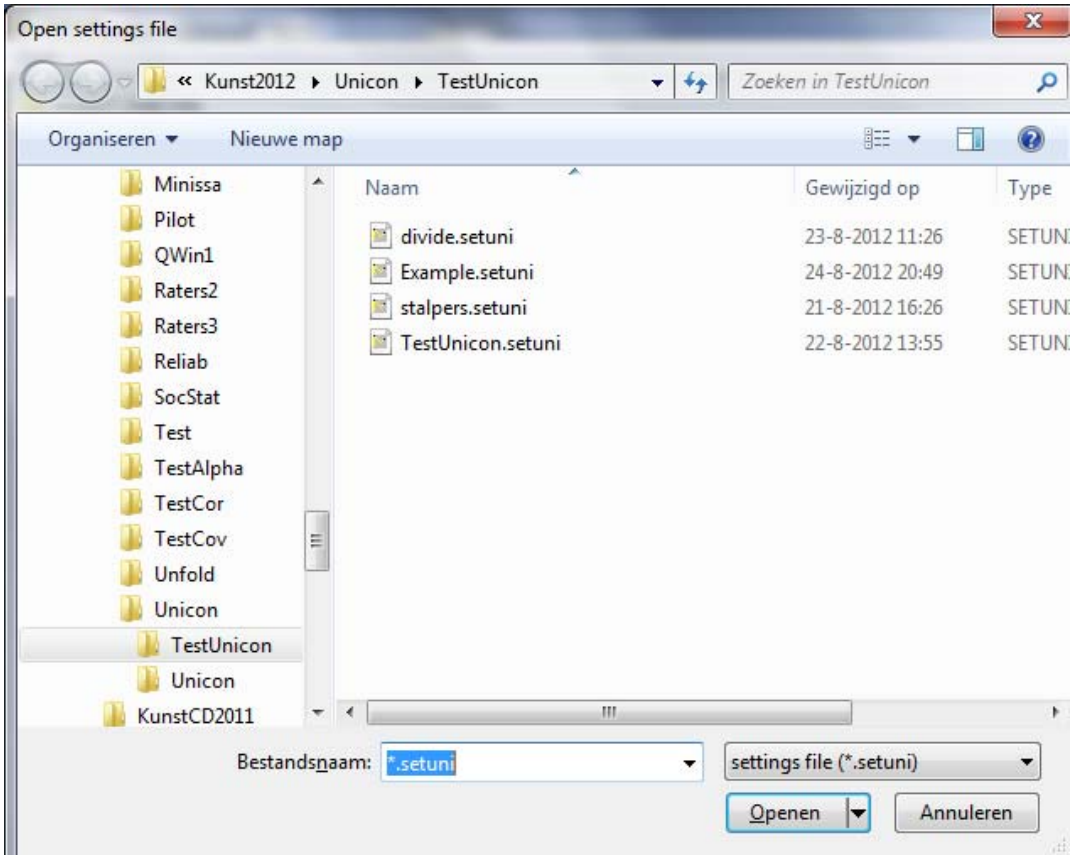

Figure 10: Opening a settings file (Dutch version of Windows).

By default settings files from *Unicon* have the extension '`.setuni`'. It may be convenient to save your current settings before collecting new ones. The program may remind you thereof. After you have collected information from a settings file, its name will be visible on the upper right part of the main window.

7.1.11 S Save settings file

If you want to save the options and specifications that you have set so far, you can type `S`Enter. Then a file-selector box will appear that allows you to specify the place and the name of the file to which the settings must be written (see figure 11).

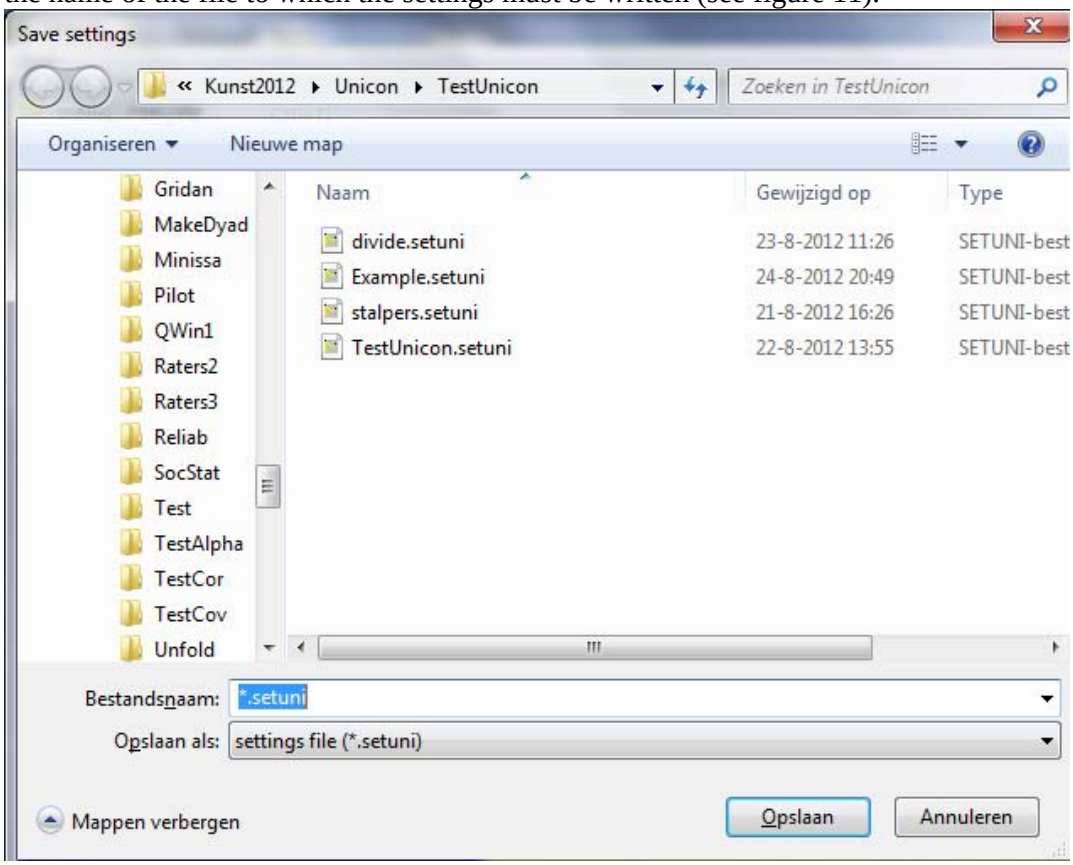

Figure 11: Saving information to a settings file.

7.1.12 X eXecute Unicon

If you have specified all options, you can type `X`Enter to start the computations. The program will check if all obligatory options are specified and if there are no inconsistencies. If everything is all right, the computations will start. If the program has been correctly installed and runs without problems, it will, when it is finished, automatically open the last (or only) listing file it has made. If it fails to do so, you can open it yourself by any text editor like *WordPad*, *Notepad* or *Word*. In order to have a nicely outlined text, you must select a small non-proportional font like `Courier New 9`.

7.1.13 Q Quit

The option Q is an emergency exit. If you choose it, *Unicon* will halt without performing any calculations and without producing any output files.

7.2 Defining factors, levels and equivalences

From the main-menu, you may define the factors by typing `F`Enter. If the number of factors is still unknown, the program will ask you to specify that first:

```
How many factors are there? (2-5):
```

If you answer the question, for instance by typing

`2`Enter

the factors-menu will open. You can avoid the question by typing the number immediately after the option code:

`F 2`Enter

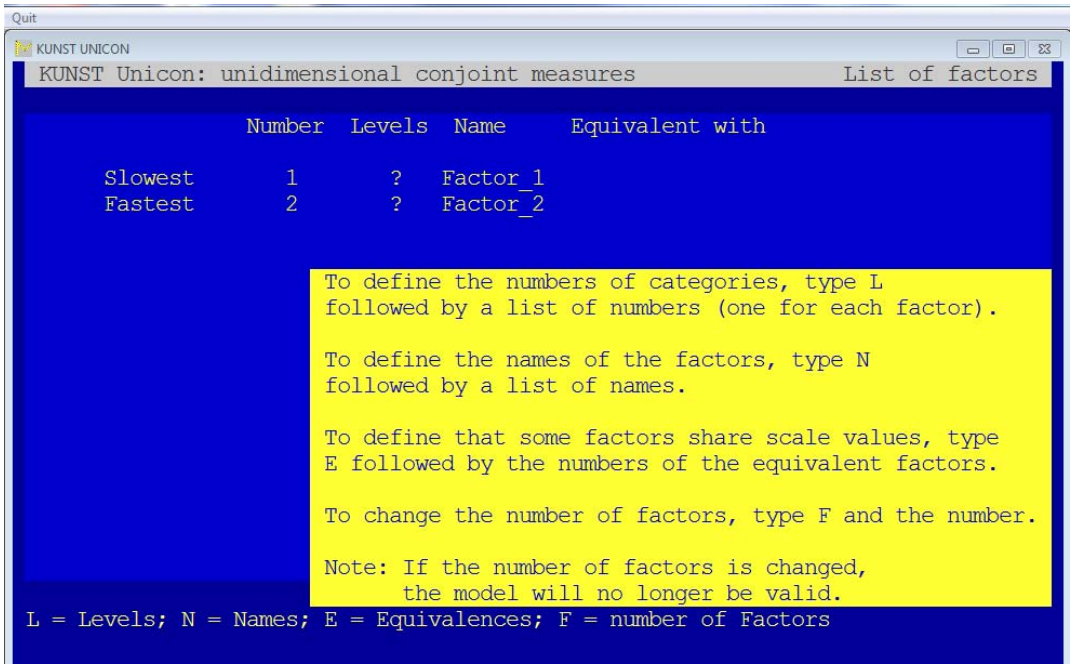

Figure 12: The factors menu

Figure 12 shows an example of the factors-menu. After you have filled out the options in this menu you must press Enter to return to the main-menu. If there is a yellow window on the screen, you must press Enter twice: once to remove the yellow window and once to leave the menu.

In the factors-menu, the option characters are shown on the bottom of the screen. We will discuss them one by one.

7.2.1 L = Levels

With this option you can define the number of levels per factor: just enter **L** followed by a list of numbers, each number being the number of levels of one factor. For example you may enter `L 5 6`Enter to define that the first factor has five levels and the second six. You can use commas to leave some levels unchanged: for instance, if you type `L 4,,2`Enter you indicate that the first factor has four levels and the third two, but that the number of levels in the second factor must stay unchanged.

7.2.2 N = Names

To give names to the factors, type **N** followed by a list of names. Names cannot be longer than 8 characters. For example, by typing `N Intensity Duration`Enter you define that the first factor has to be called "`Intensit`" and the second "`Duration`". You can use commas to skip some names: With `N ,Time`Enter you would change the name of the second factor to "`Time`", without changing the name of the first.

7.2.3 E = Equivalences

Factors may have scale values in common (see 2.3). To define these equivalences you must type **E** followed by the factor numbers whose scale values must be equivalent, for instance `E 1 2`Enter to specify that the first and the second factor must be equivalent. If equivalent factors have different numbers of levels, the equivalency holds only for the first common levels. If factors are equivalent, the screen will show so by displaying the number of the first of the equivalent factors after the later factors. To undo an equivalence type `E` followed by just one factor number.

Some examples:
If you type: `E 1 3 4`Enter the factors 1, 3 and 4 will form an equivalence group.
To undo the equivalence for factor 3, type `E 3`Enter and only factors 1 and 4 will stay equivalent.

7.2.4 F = number of Factors

By typing `F #`Enter (replacing # by a number between and 1 and 5) you can correct the number of factors.

7.3 Definition of the data

From the main-menu, you may type `D`Enter to enter the data-definition-menu. After you have filled out the options in this menu you must press Enter to return to the main-menu. If there is a yellow window on the screen, you must press Enter twice: once to remove the yellow window and once to leave the menu.

The data- definitions-menu (see figure 13) is divided into two parts: options concerning the data (the replications) and options concerning the trial configuration.

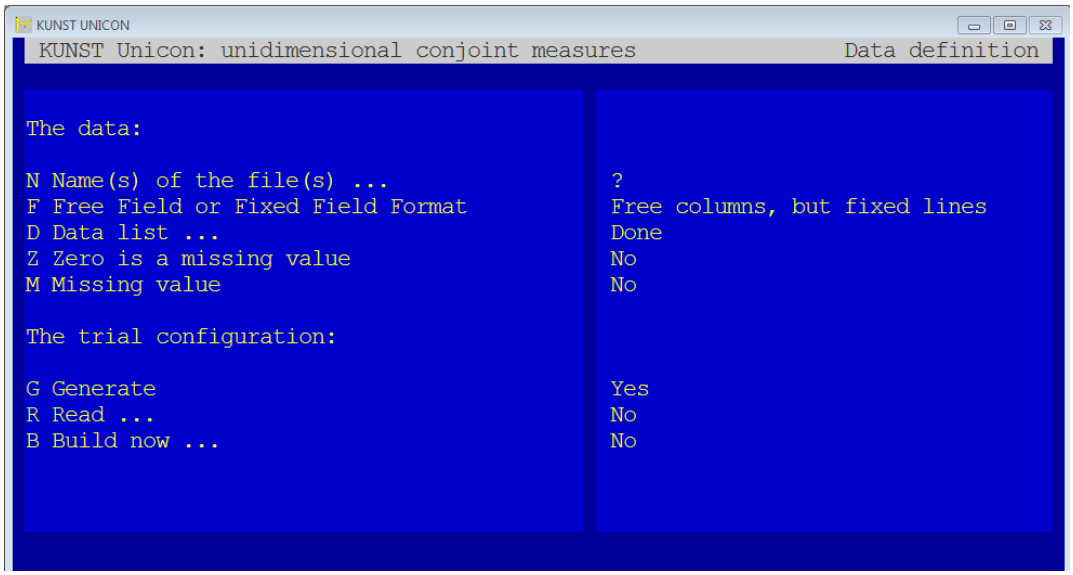

Figure 13: The data definition-menu.

For the raw data the following options can be chosen:

7.3.1 N Name(s) of the file(s)

If you choose this option a file-selector box will appear that enables you to select one or more data files. Figure 14 shows an example. Each data file must contain one or more replications.

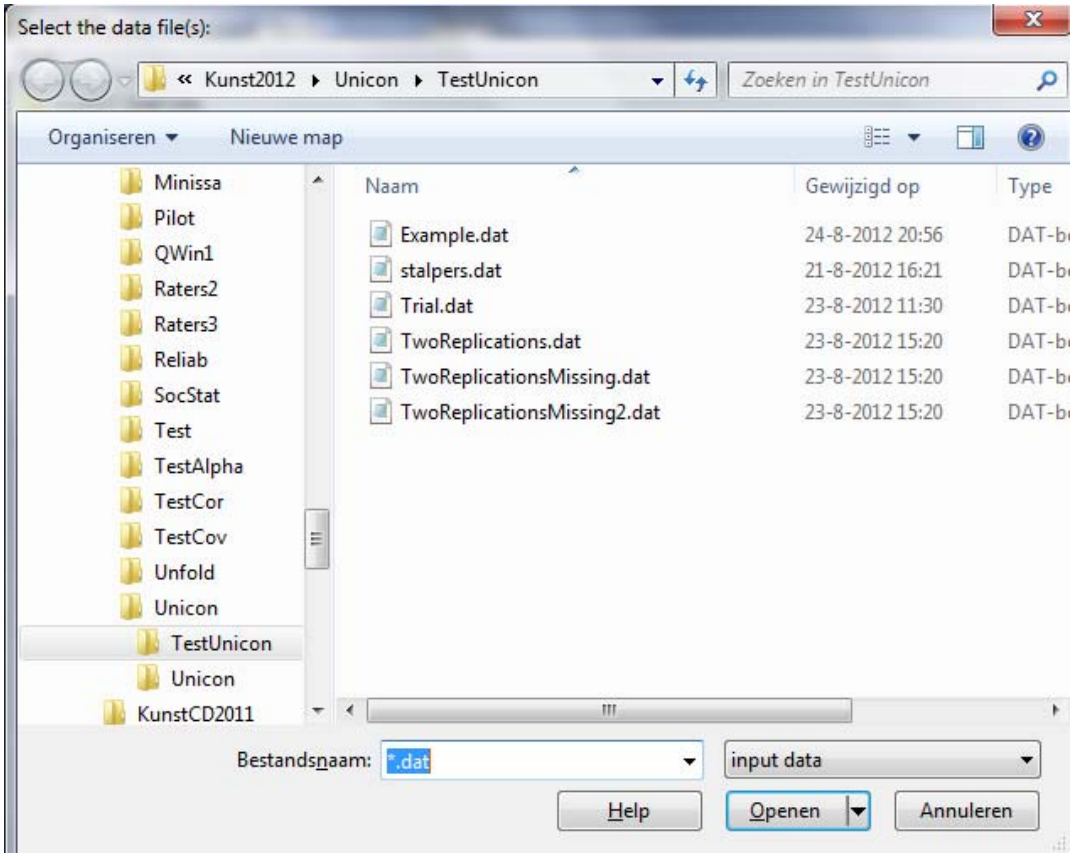

Figure 14: Selecting data files.

7.3.2 Free field or Fixed field data

The positions of the values within a replication can be organized in three different ways.

Free field format with fixed line numbers:

For each cell within the replication, the exact line number is specified. The values within the line are not bound to exact positions, but they must be separated by spaces, tabs or commas. In the file each line must contain enough numbers to fill all cells for that line.
The data must be arranged such that the line numbers are in ascending order.
If not all values are on a single line, you must use a *data list* to specify the lines (see 7.3.2.1).

```
  145 163 170 202 184 214       2 26
  337 477 557 635 637 656
  763 1035 1203 1355 1352 1595
 1302 1435 1822 1900 1927 2195
 1650 2072 2203 2545 2684 2781
```

Figure 15: Data in free field format needing the specification of line numbers.

The data in figure 15 represent a five by six replication as in the example of figure 1. At the end of the first line, the file contains two numbers that do not belong to the data for this analysis; they might represent sex and age of the replication. Therefore you must specify line numbers to indicate that the first six numbers come from the first line, but the next six from the second and so on.

Completely free format:

With completely free format neither line numbers nor positions within lines are specified. For each replication, the program will read just the number of values it needs. These values must be separated by spaces, commas, tabs or line ends. The number of lines per replication may vary. The only restriction is, that each replication must start from a new line. In order to choose this format you must activate the *datalist* and set the line number of any first stimulus to zero. See 7.3.2.2.

```
145 163 170 202 184 214 337 477
557 635 637 656 763 1035 1203
1355 1352 1595 1302 1435 1822
1900 1927 2195 1650 2072 2203
2545 2684 2781
150 160 159 200 174 203 336
479 547 624 633 653 760 1031 1198
1297 1303 1502 1432 1435 1832
1903 1929 2201 1640 2070
2213 2533
2671 2771
```

Figure 16: Data in completely free field format.

Figure 16 shows an example of a file containing two replications for a 5 by 6 design. The number of lines per replication is free, as long as there are 30 values per replication and each replication starts on a new line.

Fixed format:

If this format is used, each value within a replication has an exact line number and position. If this format is chosen, the line numbers and positions of the values must be specified in a *data list*,. See 7.3.2.2.

The data shown in figure 15 could be described by a fixed format, but that would impose unneeded restrictions on the input file.

In general, the use of free format with fixed lines is highly recommended since it is less sensitive to irregularities in the data and errors in the specifications. Moreover, if this format is used and each replication takes only one line, you do not need to fill out the data list at all.

However, in two situations a fixed format is unavoidable:

- If the data contain values that should be skipped.
- If not all values in the data are separated by spaces, commas or tabs.

Each time you type `F`Enter the choice between free field and fixed format switches.

7.3.3 D Data list...

In the data-definition-menu the option `D` leads to the data list. Such a data list may be needed to specify the input positions of the values in each replication. The exact appearance of the data list depends on the format type (see 7.3.2).

7.3.4 Free format with fixed lines

Figure 17 shows an example of a data list for data in free format with fixed lines.

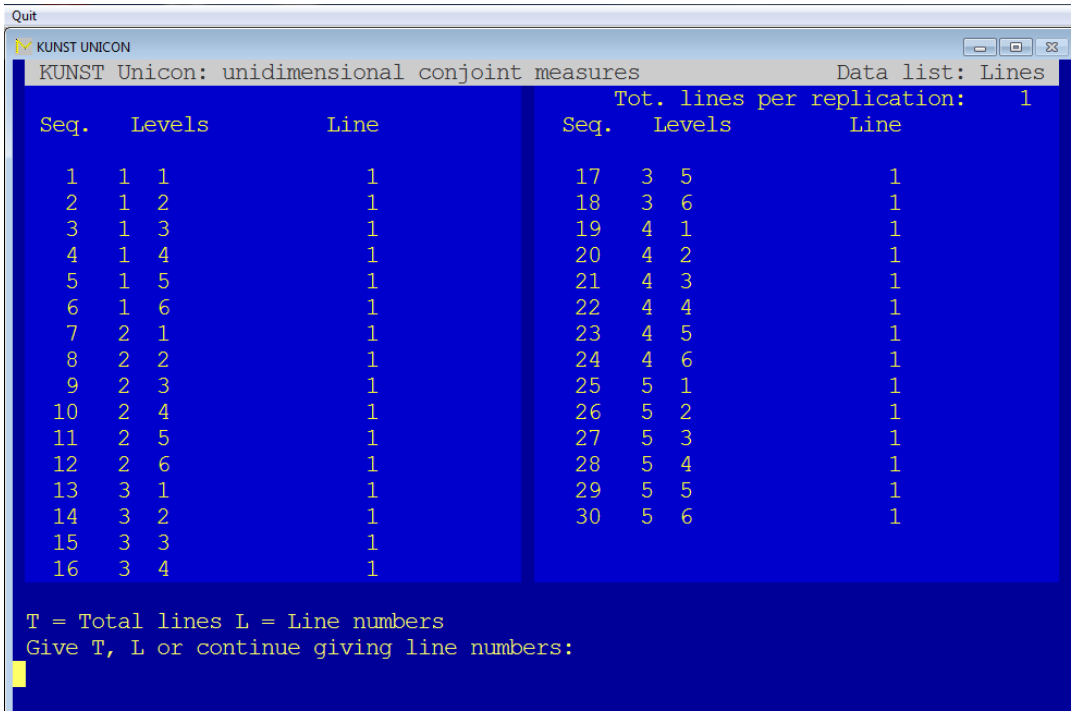

Figure 17: Example of a data list for data in free format with fixed lines.

If a free format with fixed line numbers is used and there is more than one line per replication, you must specify the line numbers. To specify these line numbers you must type the code **L** and then the sequence number of the value and its line number. You may also type a range of sequence numbers.

```
Example:   L 7-12 2Enter
```

This specification changes the line numbers for the stimuli 7 through 12 to 2. If there are more line numbers to be set, the code (L) needs not to be repeated. The line numbers must be in ascending order. Consequently, in the example, the program will automatically change the line numbers of stimuli 13 through 30 to 2, if they were still 1.

The data list shown in figure 17 forms a correct description of the data in figure 15.

The *total* number of lines per replication will automatically be adjusted to the last specified line number, but if the actual number of lines is larger than that, you can specify it explicitly by the code **T**:

Example: `T 6Enter`

This specification informs the program that each replication will occupy 6 lines.

7.3.5 Fixed format

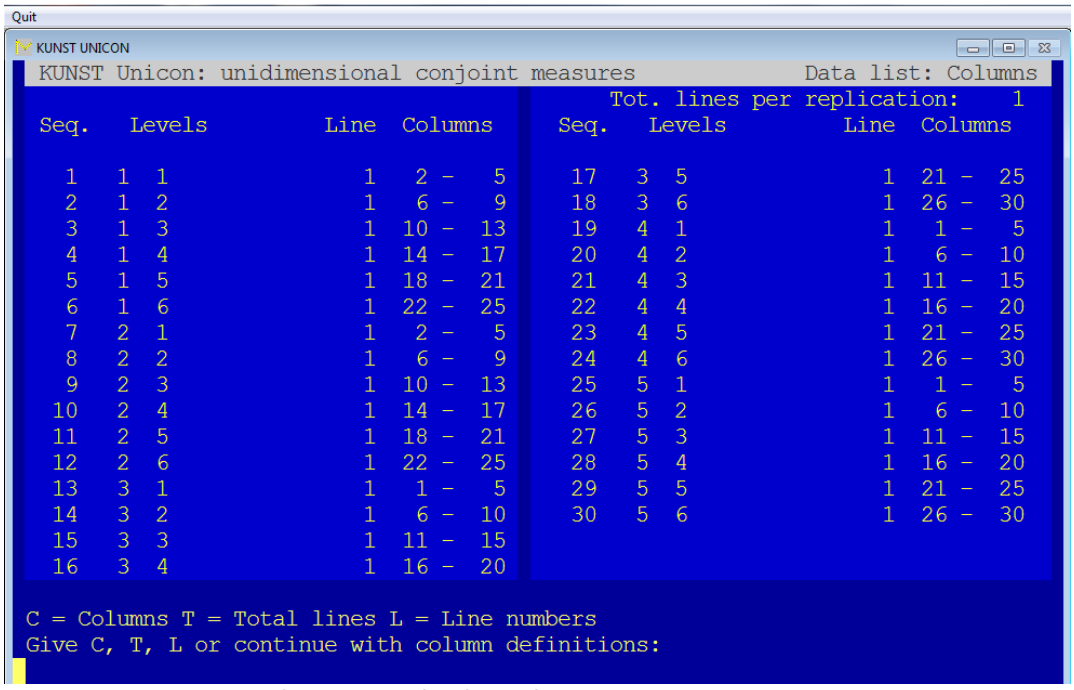

Figure 18: Example of a data list for fixed format data.

With this format type, the exact line number and position are specified for each cell in a replication. Figure 18 shows an example. The format in that figure fits the data in figure 15. The specifications could be entered as follows:

```
      L 1-6 1    7-12 2    13-18 3    19-24 4    25-30 5
      C 1-6 2-25 7-12 2-25 13-18 1-30 19-24 1-30 25-30 1-30
```

To specify line numbers you must type the code **L** and then the sequence number of a variable and its line number. You may also type a range of consecutive numbers.
Examples: `L 7-12 2`Enter
`L 13-18 3`Enter
These specifications change the line numbers for the cells 7 through 12 to 2 and the line numbers for cells 13 through 18 to 3. The line numbers must be in ascending order. Consequently, the program will automatically change the line number of stimuli 19 through 30 to 3, if they were still 1 or 2. After you have specified the line numbers for a cell or a range of cells, you may continue defining line numbers without repeating the character `L` itself.

To specify the **positions** for the cells you must use the code **C**. The use will become clear from the following list of possibilities:
- One cell in one column: specify just the cell number and the position:
  `C 1  3`Enter
  {cell 1 is in position 3}.
- One cell in more positions: specify the cell number and the range of positions:
  `C 1 2-5`Enter
  {cell 1 is in positions 2-5}.
- More cells in consecutive columns: specify the range of cell numbers and the first position:
  `C 1-10 10`Enter
  {cells 1 to 10 are in positions 10-19, each occupying one position}.
- More cells in consecutive columns with more than one position per cell: specify the range of cell numbers and the range of positions. The number of positions must be a multiple of the number of cells in the command:
  `C 1-6 2-25`Enter
  {cells 1 to 6 are in positions 2-25, each occupying four positions}

After you have specified the positions for a cell or a range of cells, you may continue defining positions without repeating the character `C` itself.

The total number of lines per replication will automatically be adjusted to the last specified line number, but if the actual number of lines is larger than that, you can specify it explicitly by the code **T**:

Example: `T 6`Enter

This specification informs the program that each replication will occupy 6 lines.

You type: `F`Enter or `B`Enter
If there are more than 32 data cells in a replication, they will not fit at once on the data-list screen. Therefore, you have the possibility to scroll **f**orward and **b**ackward with the options **F** and **B**.

7.3.6 Z Zero is a missing value

With this switch you can define whether zero is to be treated as a missing value or not.

7.3.7 M Missing value

With the option **M** you can define an upper limit for the data cells. If a cell contains a value that is equal to this upper limit or greater, its value will be treated as unknown. For example, if you type `M 99Enter` you define that the values 99 and greater are to be treated as missing values. The upper limit cannot be zero or negative.
If you do not enter a value for this option or specify zero, the option will switch to `no`.

7.3.8 The trial configuration options

For the trial configuration the options **G**, **R** and **B** can be chosen. These three options are mutually exclusive: if one of them is chosen, the others are cleared and the only way to clear one is to choose another.

G `Generate`
By this option you indicate that the trial configuration has to be generated automatically by the program.

R `Read ...`
This option is used to indicate that the trial configuration must be read from a file. The option leads to the read-trial-menu as shown in figure 19.

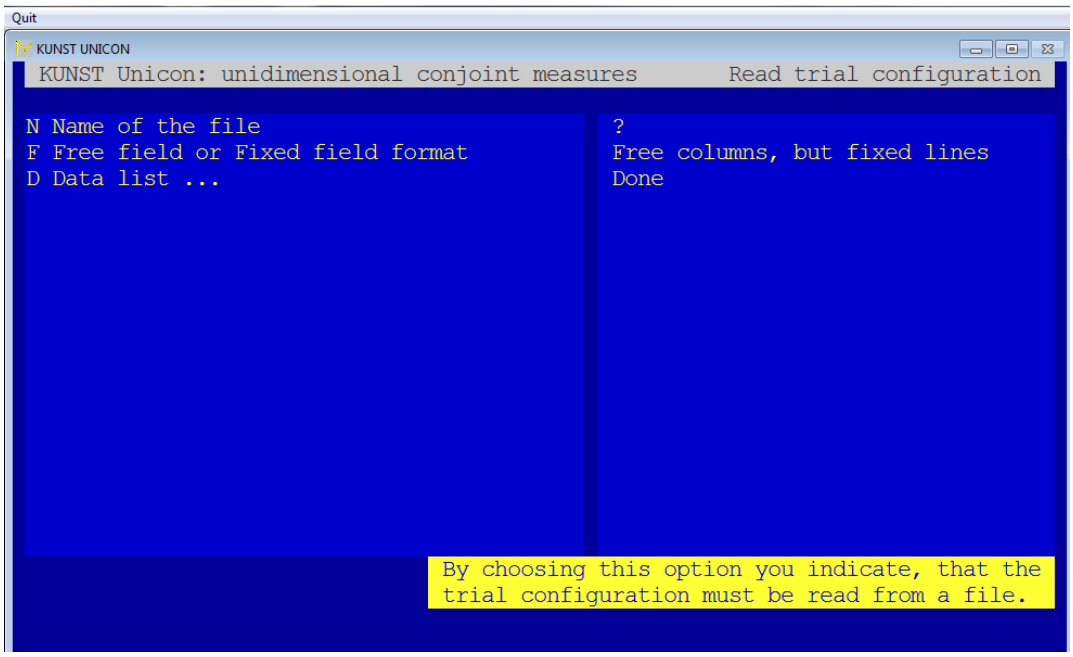

Figure 19: Reading the trial configuration from a file.

In this submenu, you must at least define the file-name (option **N**). The options **F** and **D** are the same as described for the data in 7.3.2 and 7.3.3. However, here you must define how the initial scale values have to be read from the file with the trial configuration. As an aid, the factors and their levels are shown in the data list.

If the trial configuration has to be read from the same file as the data, the trial configuration must come first.

B Build ...
This option is used to define the trial configuration immediately. The corresponding ***build***-menu is shown in figure 20. In this menu, you can define the initial scale values.

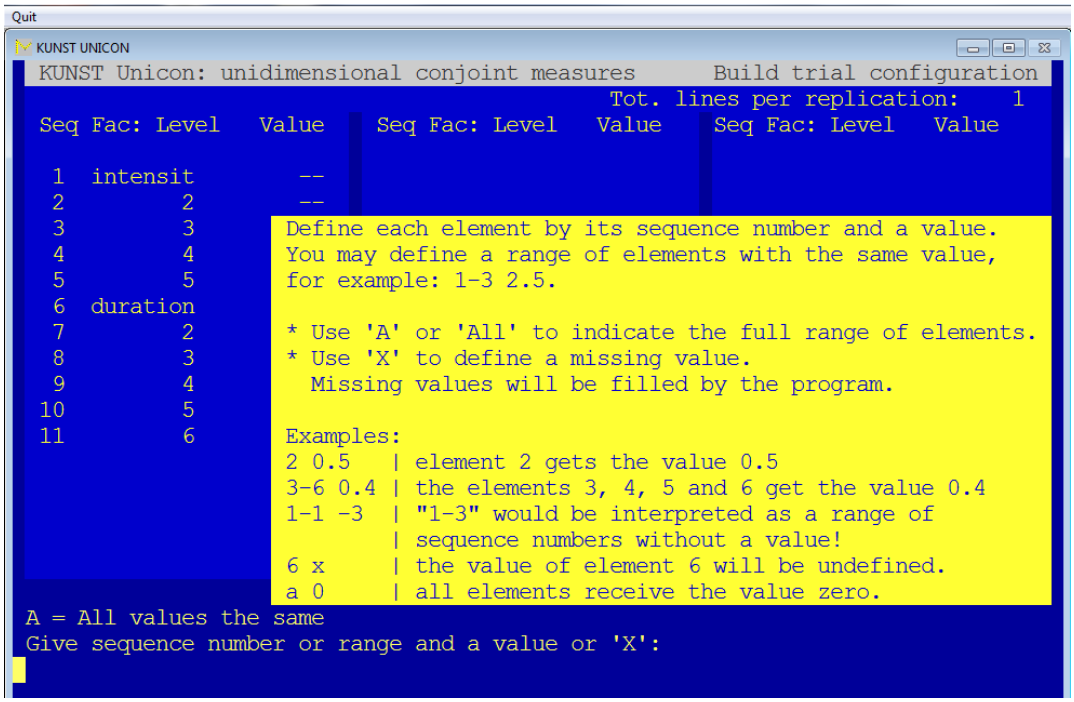

Figure 20: Building the trial configuration.

It is obvious that the number of factors and their levels must be known in advance.

To specify values for the trial configuration, you must give a sequence number (or a range of sequence numbers) followed by a value. If a value is a negative whole number, you must always use a range, for example: `1-1 -4`.

Example:
`1-3 -12.34`Enter or `5-5 -7`Enter
If you would type `5 -7`Enter the program would understand that you start giving values for the cells five through seven and complain that no value is given.

If you want to leave a value undefined, use the character `X` instead of a value. Then the value will be derived from the generated trial configuration.
If you want to define one single value for all elements in the configuration, you can type `A #`Enter where # must be replaced by the common value.

When two factors are equivalent and you assign values to one of them, the same values are given to the corresponding levels of the equivalent factor(s).

7.4 The definition of the model

In the main-menu, typing `m`Enter leads to model-menu as shown in figure 21. It allows you to define the measurement model.

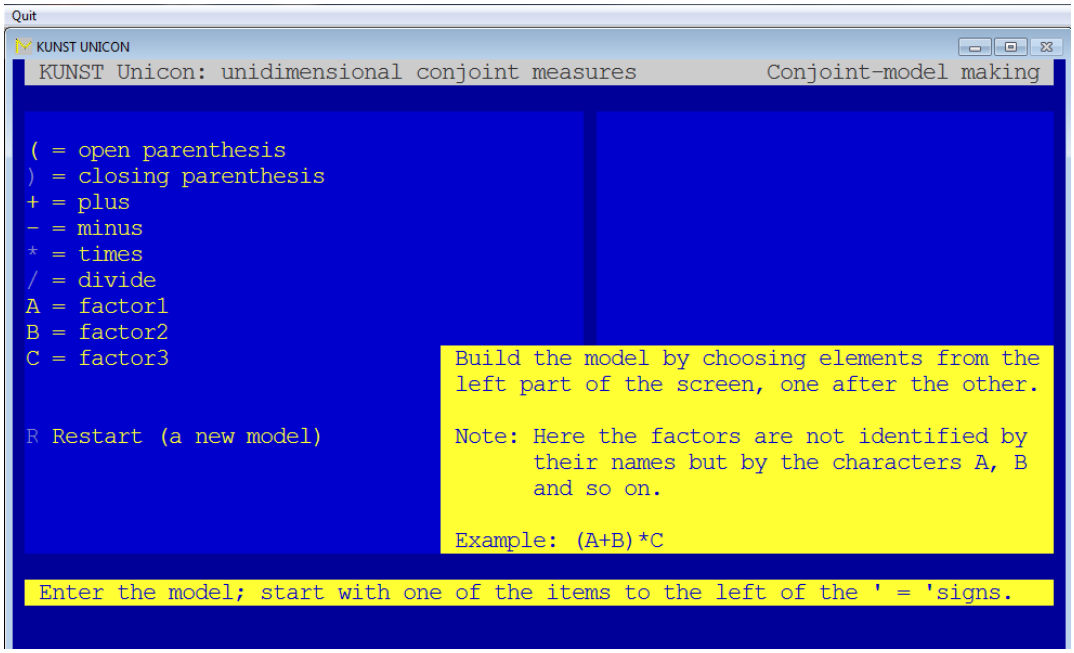

Figure 21: Example of the model-menu.

Figure 22 shows the Model-menu in the middle of the creation process.

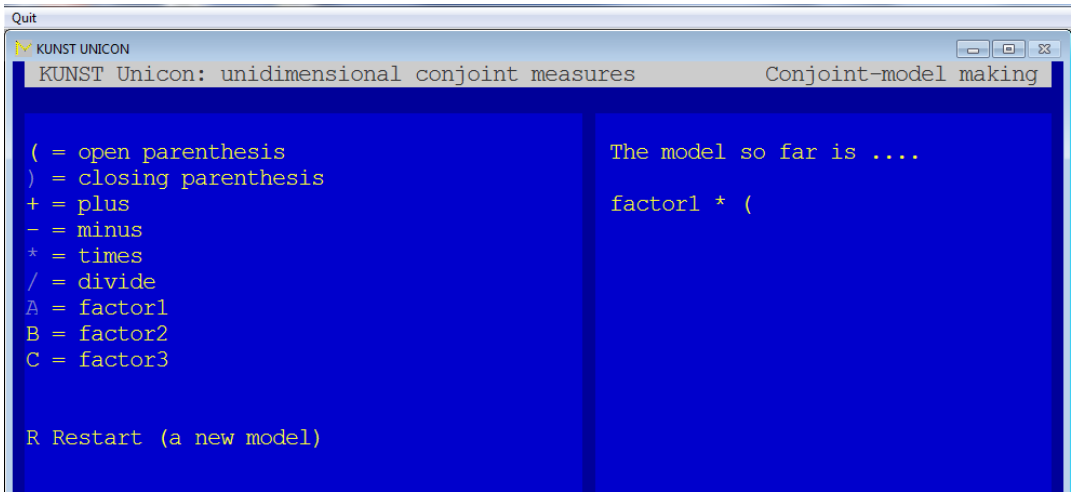

Figure 22: The model-menu in the middle of the creation process

The models supported by *Unicon* are those in which each factor is used only once and in which only the following dyadic operations are used: addition (+), subtraction (-), multiplication (∗) and division (/). The monadic operators + and - are allowed as well. The priorities of the operations are as in normal arithmetic and they may be changed by the use of parentheses. After you have used the last factor, there is no need to complete the model with closing parentheses. The program will complete the model automatically.

The *model*-menu does not allow defining an invalid model: only the elements that are acceptable (not gray) can be chosen at each stage in the process.

`R Restart new model`
If you make a mistake while defining the model, you cannot immediality correct it. In that case you must start again by typing `R`Enter .

Division (/) causes much more trouble for the algorithm than multiplication: imagine that the cell o_{ij} having the highest rank is represented by $z_{ij} = s_{1i} / s_{2j}$, where s_{1i} is positive and large and s_{2j} is positive but close to zero. If s_{2j} becomes smaller and smaller, z_{ij} increases as long as s_{2j} stays positive. On the moment that s_{2j} crosses zero, however, z_{ij} suddenly becomes the smallest z instead of the largest and the stress becomes extremely large. Although the algorithm tries to prevent such disasters by reducing its step-size, it still remains advisable to replace division by multiplication. To return to the division model, you can to invert the final scale values afterwards.

Use division only if the multiplicative approach fails or if equivalences force you to do so, as in the model `A*B+C/D`, where `B` and `D` must be equivalent.

For similar reasons it seems wise to replace multiplication (`*`) by addition (+) and division (`/`) by subtraction (`-`), if you require (or expect) that scale values of the same factor have the same sign. You can exponentiate the final scale values afterwards.

7.5 Tuning of the process

In the *main* menu, typing p`Enter` leads to a submenu as shown in figure 23. It allows you to set some parameters that influence the computations to be performed.

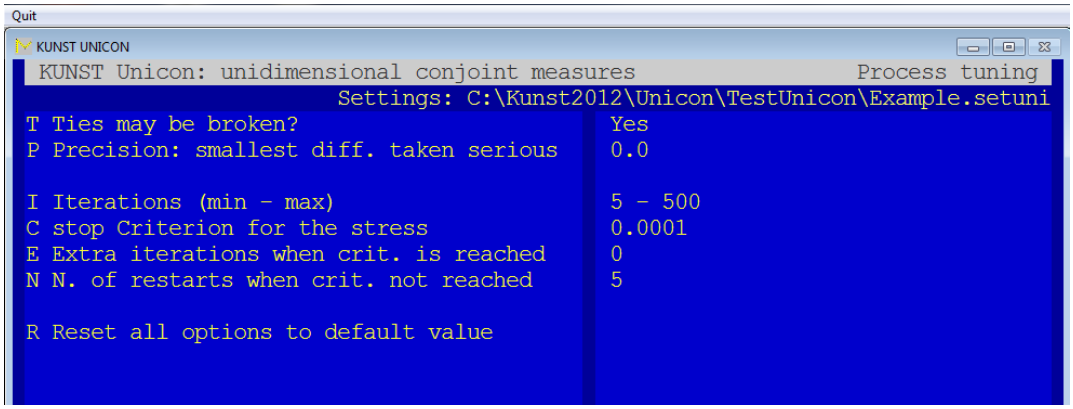

Figure 23: Process tuning

`T Ties may be broken?`
The option `T` determines whether the program must try to represent equal rankings in the data by equal distances or not. If ties may be broken the stress may be somewhat smaller than if they are not (see 0).

`P Precision: smallest diff. taken serious`
By typing `p ##Enter` one can indicate that two data values are considered to be tied if their difference is smaller than ##. There may be some ambiguity in this definition of ties: if the data are ordered there may be a chain of small differences, each smaller than the criterion. In that case the program scans the data from small to large. The first value starts a tie. All subsequent values that differ less than the criterion from the first value are added to the tie. The first value that differs too much from the starting element will act as the start of a new tie and so on (see 0).
Choose zero unless you have good reasons to give another value.

`I Iterations (min - max)`
Unicon ends its iteration process when the configuration does not improve any more (see 3.1). But the user may put some restrictions on this mechanism by defining a minimum and a maximum number of iterations to be performed. By default the minimum is 5 (do not give up too soon) and the maximum is 100. By this option one may change the values.

`C stop Criterion for the stress`
The search for a solution continues until the stress becomes lower than the value given with this option. Choosing a very small value and a large number of iterations may cause a great increase of computer time. For that reason the absolute minimum for the stress is set to 0.00001. See 3.1 for possible reasons to end the iteration process.

`E Extra iterations when stress is reached`
By typing `e#Enter` you define the number of additional iterations to be performed after the stress criterion is met.

`N N. of restarts when stress not reached`
Since the iterative steepest descent method may land in a local minimum, *Unicon* allows the user to repeat the process one or more times with a so-called *violent motion*. By typing `n#Enter` you define the maximum number of violent motions. If a solution is found with a stress less than the one defined by the option `C`, no restarts will be performed anymore. Finally the program will retain the best overall solution.

`R Reset all options to default value`
By typing `rEnter` one resets all the other options in this menu to their default values.

7.6 Additional results in the listing file

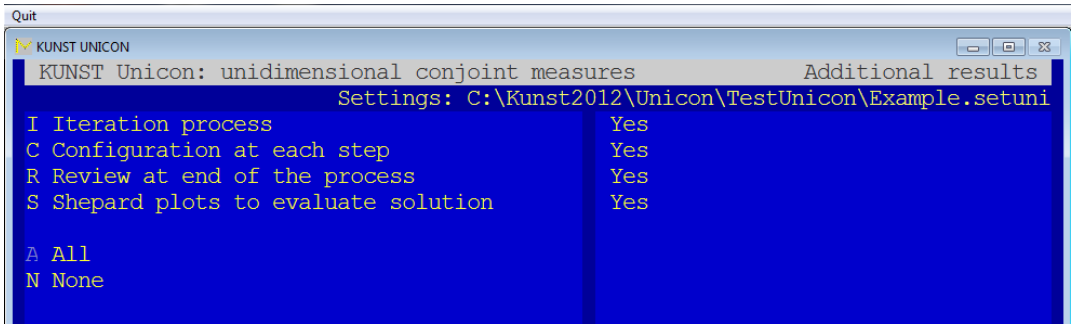

Figure 24: The additional results menu.

Entering A Enter in the *main* menu leads to the *additional results* menu (see figure 24). The options in this menu offer the probability to get additional information in the listing file besides to what is given by default.

The menu contains the following options:

`I Iteration process`
By typing i Enter you switch the option from yes to no or the other way around. If it is yes, the listing file will contain some information after each iteration step:
the magnitude of the gradient, the step-size and the stress are reported (see also 8.1). It is not advised to use this option unless one has a good reason to look into the details of the process.

`C Configuration at each step`
By typing c Enter you switch the option from yes to no or the other way around. If it is yes, the listing file will contain the configuration after each iteration step. It is only meaningful in conjunction with the option I.

`R Review at end of the process`
By typing r Enter you switch the option from yes to no or the other way around. If it is yes, the listing file will contain at the end of the process a review of the dependent scale values z, the disparities $\hat{z}$ and the contributions to the squared stress of each data element, in the order of the data within each replication.

`S Shepard plots to evaluate solution`
By typing s Enter you switch the option from yes to no or the other way around. If it is yes, the listing file will contain for each replication a so-called Shepard diagram in which z and $\hat{z}$ are plotted against the ranked data.
Note: When there are many replications, this option will substantially increase the number of lines in the listing file. In the listing file, these plots are rather coarse images build as characters in a grid. More detailed pictures will be stored in separate bitmap files.

`A All`
`N None`
If you type a Enter, options I,C,R and S will be switched to yes all together.
If you type n Enter, options I,C,R and S will be switched to no all together.

8 Results

After execution of *Unicon*, you will find one or more new files in your working directory:
- the listing file (.lst)
- if you asked for the configuration to be written in a separate file (`W` in the *main* menu) the raw output file (.out)
- possibly some bitmap files

8.1 Results in the listing file

A listing file will contain the main results of the analysis in the corresponding data file. Its precise content depends on the specifications by the user and the input data. The lines in the listing have a length of 80 characters or less. View (and print) this file with a small non-proportional font like `Courier New 9`.

- The file starts with an overview of the options as they are chosen by the user. Figure 25 shows an example.

```
 View (and print) this file with a small non-proportional font like Courier 9.
 Lines contain 80 characters or less.
 ------------------------------------------------------------------------------


 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 24-08-2012  20:57:06

     Unicon: KUNST program for UNIdimensional CONjoint Measurement
            Interactive version  0.10, September 2012

 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

 The operation and the accuracy of the program are not guaranteed.



 Unicon example
==============================================================================

   1.  Summary of the user specifications and input data.


==============================================================================

 There are 2 factors:           Factor     Name         Levels
                                   1       intensit       5
                                   2       duration       6

 The number of cells in the data matrix is 30.
 The input is supposed to contain the following cells:

 Cell   Name             Line

     1 C11                1
     2 C12                1
     ...
     ... and so on ...
     ...
```

```
    30 C56               5


 For a full row of 30 cells 5 lines will be read.

 The number of replications will be computed.
 All replications will be echoed.

 Ties in the data may lead to unequal values in the solution.
 The smallest difference to be taken seriously is 0.0.

 The model is:
   intensit * duration

 The iteration process stops when the stress is less than 0.0001.
 The number of iterations will be at least 5 and at most 500.
 After the stress criterion is met, no additional iterations will be
performed.
 The maximum number of violent motions is 5.

 In addition to the standard results, this listing will contain information
 on the following topics:
   - The iteration process
   - The configuration at each step
   - A review at the end of the iteration process
   - A Shepard plot for each replication

 The final configuration will be stored in a separate file with the name:
   C:\Kunst2012\Unicon\TestUnicon\Example.out

 The 11 scale values will be written in the following format:

   Line  1, positions  1-10: scale value  1
   Line  1, positions 11-20: scale value  2
   Line  1, positions 21-30: scale value  3
   Line  1, positions 31-40: scale value  4
   Line  1, positions 41-50: scale value  5
   Line  1, positions 51-60: scale value  6
   Line  1, positions 61-70: scale value  7
   Line  1, positions 71-80: scale value  8
   Line  2, positions  1-10: scale value  9
   Line  2, positions 11-20: scale value 10
   Line  2, positions 21-30: scale value 11

 The number of elements in the trial configuration is 11.
 The trial configuration will be generated.




 The data matrix:
 ________________

 The data will be read from the following file:
   C:\Kunst2012\Unicon\TestUnicon\Example.dat

 The data will be read now:


 Data of first replications:

 Replication 1:

  Factor     1  Factor 2 (duration)
  (intensit)

         1         2         3         4         5         6

   1   145.0     163.0     170.0     202.0     184.0     214.0
```

```
   2   337.0     477.0     557.0     635.0     637.0     656.0
   3   763.0     1035.0    1203.0    1355.0    1352.0    1595.0
   4   1302.0    1435.0    1822.0    1900.0    1927.0    2195.0
   5   1650.0    2072.0    2203.0    2545.0    2684.0    2781.0


 The number of replications is 1.
```

Figure 25: Start of a listing file.

- Then follows the trial configuration as generated by the program or given by the user. Figure 26 shows an example.

```
Unicon example
================================================================================

   2. Looking for a solution

================================================================================




 The trial configuration:
 ________________________

 Factor        Levels

 1 intensit   -0.2560     0.7990      1.367      1.830      2.265
 2 duration    0.6747     0.9641      1.019      1.215      1.189      1.387

 The stress of this trial configuration is 0.0705172.
```

Figure 26: Report of a trial configuration.

- If the details of the iteration process are requested, the listing contains at each iteration-step the iteration number, the magnitude *m* of the gradient, the step size α and the stress (S-form2), as shown in figure 27.

```
 Minimizing stress
 _________________

                       Iteration   Magnitude       Step           Stress

                               0     0.11399         0.45           0.0705172
 ------------------------------------------------------------------------------
 Configuration: (independent scale values)

 1 intensit     -0.2560     0.7990     1.3672     1.8298     2.2650
 2 duration      0.6747     0.9641     1.0192     1.2152     1.1894     1.3871

                      Step size reduced:           0.045


                               1     0.0216          0.007909        0.0189934
 ------------------------------------------------------------------------------
 Configuration: (independent scale values)

 1 intensit     -0.2515     0.7744     1.3251     1.8106     2.3273
 2 duration      0.8178     0.9627     1.0544     1.1979     1.2161     1.2979

                               2     0.02421         0.0058653       0.0169867


    ...
    ... and so on ...
    ...
```

```
------------------------------------------------------------------------------
 Configuration: (independent scale values)

 1 intensit     -0.0015     0.8057     1.4150     1.8979     2.4316
 2 duration      0.8815     1.0037     1.1450     1.2438     1.2438     1.3462


 Process ended because the stop criterion is met.

 After 46 iterations the stress is 0.0000915.
```

Figure 27: A report of the minimization process.

The updating of the configuration is performed according to the formula:

$$s_{ij} \leftarrow s_{ij} - \alpha \times s'_{ij}/m_{ij}$$

where s'_{ij} is the partial derivative of the stress with respect to the scale value s_{ij}, α is the step-size and m_{ij} is the magnitude of the gradient, defined by the following formula:

$$m_{ij} = \sqrt{\frac{\sum_{i=1}^{f}\sum_{j=1}^{l_i} s'^{2}_{ij}}{\sum_{i=1}^{f}\sum_{j=1}^{l_i} s^{2}_{ij}}}$$

The computation of the step-size α is rather complicated; it is described by Kruskal (1964b, p.120). The initial value of α is set to $0.20 + 0.05 \times r$, where r is the maximum number of "restarts" still to be performed. Usually α will decrease when the iteration process goes on.
If in one step the stress increases by more than 20% or to over 0.95, the algorithm will step back to its previous configuration and reduce its step-size by a factor of 0.1; this process of reducing the step-size may be performed up to 5 times in succession thereby reducing the step-size by a factor of 0.00001 (see Lingoes, page 393). If in the listing the iteration process is shown, the step back mechanism is reflected by a list of decreasing step-sizes immediately after each other, of which the second and following are marked by the text "step-size reduced".

- If the configuration is requested at each step (option `C` in the *additional results* menu), the current estimates of the independent scale values are given after each iteration step.
- The final solution (the stress and the configuration) is shown. The solution is also shown as a matrix of scale values of dependent and independent variables. See figure 28 for an example.

```
The final configuration:
 ________________________


 1 intensit     -0.001295   0.805753   1.415059   1.898016   2.431650
 2 duration      0.881485   1.003704   1.145031   1.243870   1.243858
1.346280



 Matrix with dependent variable and scales:
 ___________________________________________
```

Factor 1 (intensit)		Factor 2 (duration)					
		0.8815	1.0037	1.145	1.2439	1.2439	1.3463
		1	2	3	4	5	6
-0.0013	1	-0.0011	-0.0013	-0.0015	-0.0016	-0.0016	-0.0017
0.8058	2	0.7103	0.8087	0.9226	1.0023	1.0022	1.0848
1.4151	3	1.2474	1.4203	1.6203	1.7601	1.7601	1.9051
1.898	4	1.6731	1.905	2.1733	2.3609	2.3609	2.5553
2.4316	5	2.1435	2.4407	2.7843	3.0247	3.0246	3.2737

Figure 28: Report of the final solution.

- Optionally, a review is given of all cells ordered by data value within each replication, with the fitting values, the dependent scale values and the contributions to the squared stress:

$$S^2_{ij...r} = \frac{(z_{ij...r} - \hat{z}_{ij...r})^2}{\sum_i \sum_j \sum_{...} n_{ij...} (z_{ij...} - \bar{z})^2}$$

Missing data are omitted in this table.
In the case of more than one replication, the listing contains the contribution to the squared stress of each data cell over all replications:

$$S^2_{ij...} = \sum_r S^2_{ij...r}$$

Figure 29 contains an example.

Review of dependent scale values compared with ranks in the data

Replication 1

Rank nr	Disparities	Dependent values	Contribution to stress**2	Factor levels 1 inte nsit	2 dura tion
1.0	-0.001481	-0.001141	0.0000000039	1	1
2.0	-0.001481	-0.0013	0.0000000011	1	2
3.0	-0.001481	-0.001483	0.551326D-13	1	3
4.0	-0.001481	-0.001611	0.0000000006	1	5
5.0	-0.001481	-0.001611	0.0000000006	1	4
6.0	-0.001481	-0.001743	0.0000000023	1	6
7.0	0.7102593	0.7102593	0.0	2	1
8.0	0.8087374	0.8087374	0.0	2	2
9.0	0.9226125	0.9226125	0.0	2	3
10.0	1.0022472	1.002252	0.765232D-12	2	4
11.0	1.0022472	1.0022424	0.765232D-12	2	5
12.0	1.0847695	1.0847695	0.0	2	6
13.0	1.2473535	1.2473535	0.0	3	1
14.0	1.4203001	1.4203001	0.0	3	2
15.0	1.620287	1.620287	0.0	3	3
16.0	1.6730726	1.6730726	0.0	4	1
17.0	1.7601326	1.7601326	0.0	3	5
18.0	1.7601494	1.7601494	0.0	3	4
19.0	1.9050455	1.9050455	0.0	4	2
20.0	1.9050664	1.9050664	0.0	3	6
21.0	2.143463	2.143463	0.0	5	1
22.0	2.1732875	2.1732875	0.0	4	3
23.0	2.3608734	2.3608846	0.424609D-11	4	4
24.0	2.3608734	2.3608621	0.424609D-11	4	5
25.0	2.440656	2.440656	0.0	5	2
26.0	2.5552614	2.5552614	0.0	4	6
27.0	2.7843152	2.7843152	0.0	5	3
28.0	3.0246415	3.0246559	0.696933D-11	5	4
29.0	3.0246415	3.024627	0.696933D-11	5	5
30.0	3.2736824	3.2736824	0.0	5	6

The contribution of replication 1 to the squared stress is 0.0000000084.

Figure 29: Review of the dependent scale values.

- The number of equivalence classes (distinct values) in the data, in the dependent scale values and in the fitting values. See figure 30 for an example.

Number of distinct values: (degeneration shown by a smaller number for dependent scale values and disparities)

```
    30 in data                    (equiv. values differ no more than 0.0)
    22 in dependent scale values (equiv. values differ no more than 0.0005)
    21 in disparities            (equiv. values differ no more than 0.0005)
```

Figure 30: Report of the numbers of equivalence classes.

- The process of building equivalence classes in the data is described in 2.6. In the same way equivalence classes are built for the dependent scale values z and the fitting values $\hat{z}$ but here the criterion value for fusion is c:

$$c = 0.0005 \times \frac{\sum_i \sum_j \sum_{\ldots} (z_{ij\ldots} - \bar{z})^2}{\sum_i \sum_j \sum_{\ldots} n_{ij\ldots}}$$

- Optionally a Shepard diagram for each replication is shown. An example is shown in figure 31.
  The ranks of the data are on the horizontal axis and the dependent scale values z and the fitting values $\hat{z}$ both on the vertical axis. The symbol used for the z-values is the character X and the symbol for the $\hat{z}$-values is a minus sign (-).
  When the two symbols share the same location an asterisk (*) is shown.

```
Unicon example

================================================================================

   3. Shepard plot(s)

================================================================================
 The following Shepard plot(s) give an impression of the fit.
 The dependent scale values should increase.
 The disparities stay as close as possible to the dependent scale values,
     but they are not allowed to go down.
 Trivialisation is shown by a series of equal scale values.


  Shepard diagram for replication 1
  _________________________________

 Horizontal: the data
 Vertical  : - = disparity; X = dependent scale value; * = both at same point
         .+.........+.........+.........+.........+.........+.........+.
   3.274 :                                                             *:
   3.196 :                                                              :
   3.118 :                                                              :
   3.040 :                                                          * *  :
   2.962 -                                                              -
   2.884 :                                                              :
   2.806 :                                                       *      :
   2.728 :                                                              :
   2.650 :                                                              :
   2.572 -                                                    *         -
   2.494 :                                                              :
   2.416 :                                                 *            :
   2.338 :                                              * *             :
   2.260 :                                                              :
   2.182 -                                           * *                -
   2.104 :                                                              :
   2.026 :                                                              :
   1.948 :                                                              :
   1.870 :                                           * *                :
   1.792 -                                        *  *                  -
   1.714 :                                                              :
   1.636 :                                     * *                      :
   1.558 :                                                              :
   1.480 :                                                              :
   1.402 -                                  *                           -
   1.324 :                                                              :
   1.246 :                                *                             :
   1.168 :                                                              :
   1.090 :                             *                                :
   1.012 -                          * *                                 -
   0.934 :                       *                                      :
   0.856 :                                                              :
   0.778 :                    *                                         :
   0.700 :                 *                                            :
   0.622 -                                                              -
   0.544 :                                                              :
   0.466 :                                                              :
   0.388 :
   0.310 :                                                             :
   0.232 -                                                             -
   0.154 :                                                             :
   0.076 :                                                             :
  -0.002 :  * *  * * * *                                               :
         .+.........+.........+.........+.........+.........+.........+.
        0.00       5.00      10.00      15.00      20.00      25.00      30.00

 This Shepard plot is also given in file:
 C:\Kunst2012\Unicon\TestUnicon\ExampleS.bmp
```

Figure 31: A Shepard plot.

In a Shepard plot degeneration shows by dependent scale values on the same horizontal position. Violations of the model show as a descending sequence of dependent scale values.

8.2 Results in the plot file

If Shepard plots are given in the listing file, these plots are also reported in bitmap files. Figure 4 shows the bitmap corresponding to the Shepard diagram in figure 31.

8.3 Results in the raw output file

Optionally the final configuration, i.e. the scale values, is written to the raw output file. The values are written as a continuous list of numbers, eight on a line, each using 10 positions. All numbers use 10 positions and have four digits after the decimal point. An example is given in figure 32.

```
   -0.0013    0.8058    1.4151    1.8980    2.4316    0.8815    1.0037    1.1450
    1.2439    1.2439    1.3463
```

Figure 32: Content of a raw output file.

9 Literature

Bezembinder, Thom.G.G., *Van rangorde naar continuum, een verhandeling over data-structuren in de psychologie*, Van Lochum Slaterus, Deventer, pp.113-147, 1970.

Gustafsson Anders, Herrmann Andreas, Huber Frank (Eds), *Conjoint measurement: methods and applications*, 2nd ed, Springer, Berlin, 2001.

Krantz, D.H. and Tversky, A., *Conjoint measurement analysis of composition rules in psychology*, Psychological review, vol.78, no 2, pp.151-169, 1971.

Kruskal, J.B., *Nonmetric multidimensional scaling: a numerical method*, Psychometrika, Vol 29, pp. 45-129, 1964a.

Kruskal, J.B., *Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis*, Psychometrika, Vol 29, pp. 1-27, 1964b.

Lingoes, J.C., Roskam, E.E.Ch.I., and Borg, I., (Eds), *Geometric representations of relational data*, 2nd ed, Ann Arbor: Mathesis Press, 1979.

Luce, R.D. & Tukey, J.W., *Simultaneous conjoint measurement: a new type of fundamental measurement*, Journal of mathematical psychology, pp.1-27, 1964.

Maas Arne, *The use of conjoint measurement in medical decision making: a theoretical and practical contribution*, Nijmegen Institute for Cognition and Information, University of Nijmegen, 1992

Roskam, E.E.Ch.I., *Unidimensional conjoint measurement (UNICON) for multi faceted design*, Dept. of Mathematical Psychology, University of Nijmegen, The Netherlands, 1974.

Roskam, E.E.Ch.I., *Nonmetric data analysis: general methodology and technique with brief description of mini-programs*, Internal Report 75-MA-13, Dept. of Mathematical Psychology, University of Nijmegen, The Netherlands, 1975.

10 Index